

BAB II

TINJAUAN PUSTAKA

2.1. Transmission Control Protocol/Internet Protocol

TCP/IP (singkatan dari *Transmission Control Protocol/Internet Protocol*) adalah standar komunikasi data yang digunakan oleh komunitas *internet* dalam proses tukar-menukar data dari satu komputer ke komputer lain di dalam jaringan *Internet*. Protokol ini tidaklah dapat berdiri sendiri, karena memang protokol ini berupa kumpulan protokol (*protocol suite*). Protokol ini juga merupakan protokol yang paling banyak digunakan saat ini. Data tersebut diimplementasikan dalam bentuk perangkat lunak (*Software*) di sistem operasi. Istilah yang diberikan kepada perangkat lunak ini adalah *TCP/IP stack* (Wikipedia, 2009).

Protokol *TCP/IP* dikembangkan pada akhir dekade 1970-an hingga awal 1980-an sebagai sebuah protokol standar untuk menghubungkan komputer-komputer dan jaringan untuk membentuk sebuah jaringan yang luas (*WAN*). *TCP/IP* merupakan sebuah standar jaringan terbuka yang bersifat independen terhadap mekanisme *transport* jaringan fisik yang digunakan, sehingga dapat digunakan di mana saja. Protokol ini menggunakan skema pengalamatan yang sederhana yang disebut sebagai alamat *IP* (*IP Address*) yang mengizinkan hingga beberapa ratus juta komputer untuk dapat saling berhubungan satu sama lainnya di *Internet*. Protokol ini juga bersifat *routable* yang berarti protokol ini cocok untuk menghubungkan sistem-sistem berbeda (seperti *Microsoft Windows* dan keluarga *UNIX*) untuk membentuk jaringan yang heterogen.

Protokol *TCP/IP* selalu berevolusi seiring dengan waktu, mengingat semakin banyaknya kebutuhan terhadap jaringan komputer dan *Internet*. Pengembangan ini dilakukan oleh beberapa badan, seperti halnya *Internet Society (ISOC)*, *Internet Architecture Board (IAB)*, dan *Internet Engineering Task Force (IETF)*. Macam-macam protokol yang berjalan di atas *TCP/IP*, skema pengalamatan, dan konsep *TCP/IP* didefinisikan dalam dokumen yang disebut sebagai *Request for Comments (RFC)* yang dikeluarkan oleh *IETF*.

2.1.1. Arsitektur

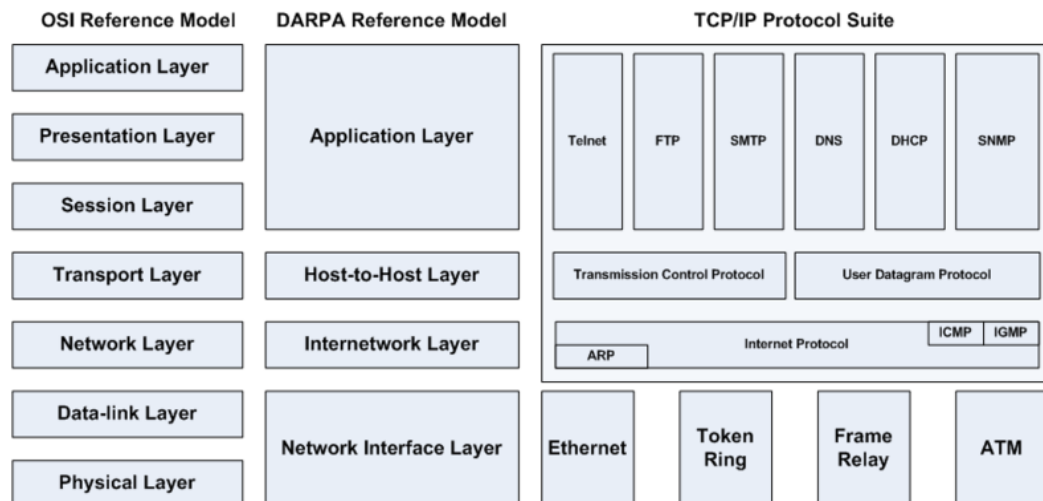
Arsitektur *TCP/IP* tidaklah berbasis model referensi tujuh lapis *OSI*, tetapi menggunakan model referensi *DARPA*. Seperti diperlihatkan dalam gambar 2.1, *TCP/IP* mengimplementasikan arsitektur berlapis yang terdiri atas empat lapis. Empat lapis ini, dapat dipetakan (meski tidak secara langsung) terhadap model referensi *OSI*. Empat lapis ini, kadang-kadang disebut sebagai *DARPA Model*, *Internet Model*, atau *DoD Model*, mengingat *TCP/IP* merupakan protokol yang awalnya dikembangkan dari proyek *ARPANET* yang dimulai oleh Departemen Pertahanan Amerika Serikat.

Setiap lapisan yang dimiliki oleh kumpulan protokol (*protocol suite*) *TCP/IP* diasosiasikan dengan protokolnya masing-masing. Protokol utama dalam protokol *TCP/IP* adalah sebagai berikut :

1. Protokol lapisan aplikasi : bertanggung jawab untuk menyediakan akses kepada aplikasi terhadap layanan jaringan *TCP/IP*. Protokol ini mencakup protokol *Dynamic Host Configuration Protocol (DHCP)*, *Domain Name System (DNS)*, *Hypertext Transfer Protocol (HTTP)*, *File Transfer*

Protocol (FTP), Telnet, Simple Mail Transfer Protocol (SMTP), Simple Network Management Protocol (SNMP), dan masih banyak protokol lainnya. Dalam beberapa implementasi stack protokol, seperti halnya Microsoft TCP/IP, protokol-protokol lapisan aplikasi berinteraksi dengan menggunakan antarmuka Windows Sockets (Winsock) atau NetBIOS over TCP/IP (NetBT).

2. Protokol lapisan antar-*host* : berguna untuk membuat komunikasi menggunakan sesi koneksi yang bersifat *connection-oriented* atau *broadcast* yang bersifat *connectionless*. Protokol dalam lapisan ini adalah *Transmission Control Protocol (TCP)* dan *User Datagram Protocol (UDP)*.
3. Protokol lapisan *internetwork* : bertanggung jawab untuk melakukan pemetaan (*routing*) dan enkapsulasi paket-paket data jaringan menjadi paket-paket *IP*. Protokol yang bekerja dalam lapisan ini adalah *Internet Protocol, Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), dan Internet Group Management Protocol (IGMP)*.
4. Protokol lapisan antar muka jaringan : bertanggung jawab untuk meletakkan *frame-frame* jaringan di atas media jaringan yang digunakan. *TCP/IP* dapat bekerja dengan banyak teknologi *transport*, mulai dari teknologi *transport* dalam *LAN* (seperti halnya *Ethernet* dan *Token Ring*), *MAN* dan *WAN* (seperti halnya *dial-up modem* yang berjalan di atas *Public Switched Telephone Network (PSTN), Integrated Services Digital Network (ISDN)*, serta *Asynchronous Transfer Mode (ATM)*.



Gambar 2.1. Arsitektur *TCP/IP* diperbandingkan dengan *DARPA Reference*

Model dan OSI Reference Model

(wikipedia online)

2.1.2. Pengalamatan

Protokol *TCP/IP* menggunakan dua buah skema pengalamatan yang dapat digunakan untuk mengidentifikasi sebuah komputer dalam sebuah jaringan atau jaringan dalam sebuah *inter network*, yakni sebagai berikut:

1. **Pengalamatan *IP*** : yang berupa alamat logis yang terdiri atas 32-bit (empat oktet berukuran 8-bit) yang umumnya ditulis dalam format `www.xxx.yyy.zzz`. Dengan menggunakan *subnet mask* yang diasosiasikan dengannya, sebuah alamat *IP* pun dapat dibagi menjadi dua bagian, yakni *Network Identifier (NetID)* yang dapat mengidentifikasi jaringan lokal dalam sebuah *internetwork* dan *Host identifier (HostID)* yang dapat mengidentifikasi *host* dalam jaringan tersebut. Sebagai contoh, alamat 205.116.008.044 dapat dibagi dengan menggunakan *subnet mask* 255.255.255.000 ke dalam *Network ID* 205.116.008.000 dan *Host ID* 44.

Alamat *IP* merupakan kewajiban yang harus ditetapkan untuk sebuah *host*, yang dapat dilakukan secara manual (statis) atau menggunakan *Dynamic Host Configuration Protocol (DHCP)* (dinamis).

2. ***Fully Qualified Domain Name (FQDN)*** : Alamat ini merupakan alamat yang direpresentasikan dalam nama *alfa numerik* yang diekspresikan dalam bentuk *<nama_host>.<nama_domain>*, di mana *<nama_domain>* mengidentifikasi jaringan di mana sebuah komputer berada, dan *<nama_host>* mengidentifikasi sebuah komputer dalam jaringan. Pengalamatan *FQDN* digunakan oleh skema penamaan *Domain Name System (DNS)*. Sebagai contoh, alamat *FQDN* *id.wikipedia.org* merepresentasikan sebuah *host* dengan nama "*id*" yang terdapat di dalam *domain* jaringan "*wikipedia.org*". Nama *domain* *wikipedia.org* merupakan *second-level domain* yang terdaftar di dalam *top-level domain* *.org*, yang terdaftar dalam *root DNS*, yang memiliki nama "." (titik). Penggunaan *FQDN* lebih bersahabat dan lebih mudah diingat dari pada menggunakan alamat *IP*. Akan tetapi, dalam *TCP/IP*, agar komunikasi dapat berjalan, *FQDN* harus diterjemahkan terlebih dahulu (proses penerjemahan ini disebut sebagai resolusi nama) ke dalam alamat *IP* dengan menggunakan server yang menjalankan *DNS*, yang disebut dengan *Name Server* atau dengan menggunakan berkas *hosts* (*/etc/hosts* atau *%systemroot%\system32\drivers\etc\hosts*) yang disimpan di dalam mesin yang bersangkutan.

2.1.3 Layanan

Berikut ini adalah layanan tradisional yang dapat berjalan di atas protokol *TCP/IP*:

1. Pengiriman berkas (*file transfer*).

File Transfer Protocol (FTP) memungkinkan pengguna komputer yang satu untuk dapat mengirim ataupun menerima berkas ke sebuah *host* di dalam jaringan. Metode *otentikasi* yang digunakannya adalah penggunaan nama pengguna (*user name*) dan (*password*), meskipun banyak juga *FTP* yang dapat diakses secara anonim (*anonymous*), alias tidak memiliki kata kunci (*password*).

2. ***Remote login.*** *Network terminal Protocol (telnet)* memungkinkan pengguna komputer dapat melakukan *login* ke dalam suatu komputer di dalam suatu jaringan secara jarak jauh. Jadi hal ini berarti bahwa pengguna menggunakan komputernya sebagai perpanjangan tangan dari komputer jaringan tersebut. *Computer mail*. Digunakan untuk menerapkan sistem surat elektronik.

3. ***Network File System (NFS).*** Pelayanan akses berkas-berkas yang dapat diakses dari jarak jauh yang memungkinkan klien-klien untuk mengakses berkas pada komputer jaringan, seolah-olah berkas tersebut disimpan secara lokal.

4. ***Remote Execution.*** Memungkinkan pengguna komputer untuk menjalankan suatu program tertentu di dalam komputer yang berbeda. Biasanya berguna jika pengguna menggunakan komputer yang terbatas, sedangkan ia memerlukan sumber yang banyak dalam suatu sistem

komputer. Ada beberapa jenis *remote execution*, ada yang berupa perintah-perintah dasar saja, yaitu yang dapat dijalankan dalam sistem komputer yang sama dan ada pula yg menggunakan sistem *Remote Procedure Call (RPC)*, yang memungkinkan program untuk memanggil *subrutin* yang akan dijalankan di sistem komputer yg berbeda. (sebagai contoh dalam *Berkeley UNIX* ada perintah *rsh* dan *rexec*.)

5. **Name server** yang berguna sebagai penyimpanan basis data nama *host* yang digunakan pada *Internet*.

2.2. RAID (*Redundant Array of Independent Drive/Disk*)

RAID adalah kependekan dari *Redundant Array of Independent Drive/Disk*. Ada juga yang menyebutnya sebagai kependekan dari *Redundant Array of Inexpensive Drive/Disk*. Secara sederhana, RAID bisa diartikan sebagai cara menyimpan data pada beberapa *harddisk*. Dengan begini, kinerja PC bisa meningkat. Selain itu, salinan data juga bisa dijadikan *backup*.

Implementasi RAID membutuhkan minimal 2 *harddisk*. Ketika RAID digunakan, sistem operasi akan membaca kedua *harddisk* sebagai 1 *harddisk*. Jadi, meskipun ada 2 *harddisk*, *drive* yang tampak pada *Windows Explorer* hanya 1 (C saja, misalnya). Sebagai perbandingan, kalau RAID tidak digunakan, *drive* pada *Windows Explorer* muncul C dan D. Setiap *drive* untuk 1 *harddisk*.

RAID menggunakan teknik *stripping*, yang membuat partisi pada ruang dengan ukuran mulai dari 512 *byte* hingga ke beberapa *megabyte*. Tiap partisi itu mengandung pecahan data yang akan dibaca bersamaan untuk mempercepat pembacaan data.

RAID memiliki beberapa level, RAID-0 sampai RAID-7 plus RAID-10 dan beberapa RAID kombinasi. Setiap level RAID memiliki fungsi yang berbeda. Penjelasannya ada di tabel level RAID.

Selain RAID yang ada di tabel, RAID punya beberapa level lagi. Misalnya Level 10 yang artinya kombinasi antara RAID-0 dan RAID-1. Ada juga RAID-50 yang merupakan kombinasi antara RAID-5 dan RAID-0. Kombinasi ini mengawinkan fungsi antara kedua RAID.

2.2.1. Level pada RAID

RAID terbagi menjadi level, yakni sebagai berikut :

1. **RAID 0:** Juga dikenal dengan modus *stripping*. Membutuhkan minimal 2 *harddisk*. Sistemnya adalah menggabungkan kapasitas dari beberapa *harddisk*. Sehingga secara logikal hanya "terlihat" sebuah *harddisk* dengan kapasitas yang besar (jumlah kapasitas keseluruhan *harddisk*).

Pada awalnya, RAID-0 digunakan untuk membentuk sebuah partisi yang sangat besar dari beberapa *harddisk* dengan biaya yang efisien.

Misalnya:

Kita membutuhkan suatu partisi dengan ukuran 500GB. Harga sebuah *harddisk* berukuran 100GB adalah Rp.500.000,- sedangkan harga *harddisk* berukuran 500GB adalah Rp.5.000.000,-. Nah, kita dapat membentuk suatu partisi berukuran 500GB dari 5 unit *harddisk* berukuran 100GB dengan menggunakan RAID-0. Tentunya skenario ini lebih murah karena memakan biaya lebih murah: $5 \times \text{Rp.500.000,-} = \text{Rp.2.500.000,-}$.

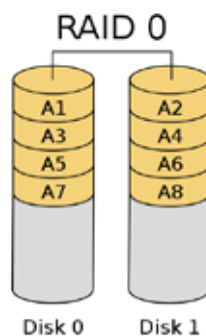
Lebih murah daripada harus membeli *harddisk* yang berukuran 500GB. Itulah kenapa pada awalnya disebut *redundant array of inexpensive disk*.

Contoh lain:

Pada saat ini ukuran *harddisk* yang tersedia di pasaran adalah 500GB, sedangkan kita membutuhkan suatu partisi dengan ukuran 2TB. Nah, kita dapat membeli 4 unit *harddisk* berkapasitas 500GB dan mengkonfigurasinya dengan RAID-0, sehingga kita dapat memiliki suatu partisi berukuran 2TB tanpa harus menunggu *harddisk* dengan kapasitas sebesar itu tersedia dipasar.

Data yang ditulis pada *harddisk-harddisk* tersebut terbagi-bagi menjadi fragmen-fragmen. Dimana fragmen-fragmen tersebut disebar di seluruh *harddisk*. Sehingga, jika salah satu *harddisk* mengalami kerusakan fisik, maka data tidak dapat dibaca sama sekali.

Namun ada keuntungan dengan adanya fragmen-fragmen ini: kecepatan. Data bisa diakses lebih cepat dengan RAID-0, karena saat komputer membaca sebuah fragmen di satu *harddisk*, komputer juga dapat membaca fragmen lain di *harddisk* lainnya.



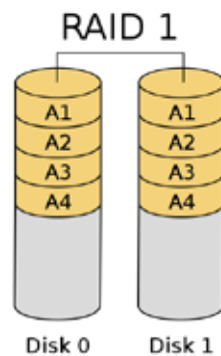
Gambar 2.1 RAID 0

(Wikipedia online)

2. **RAID 1:** Biasa disebut dengan modus *mirroring*. Membutuhkan minimal 2 *harddisk*. Sistemnya adalah menyalin isi sebuah *harddisk* ke *harddisk* lain dengan tujuan: jika salah satu *harddisk* rusak secara fisik, maka data tetap dapat diakses dari *harddisk* lainnya.

Contoh:

Sebuah *server* memiliki 2 unit *harddisk* yang berkapasitas masing-masing 80GB dan dikonfigurasi RAID 1. Setelah beberapa tahun, salah satu *harddisk*nya mengalami kerusakan fisik. Namun data pada *harddisk* lainnya masih dapat dibaca, sehingga data masih dapat diselamatkan selama bukan semua *harddisk* yang mengalami kerusakan fisik secara bersamaan.



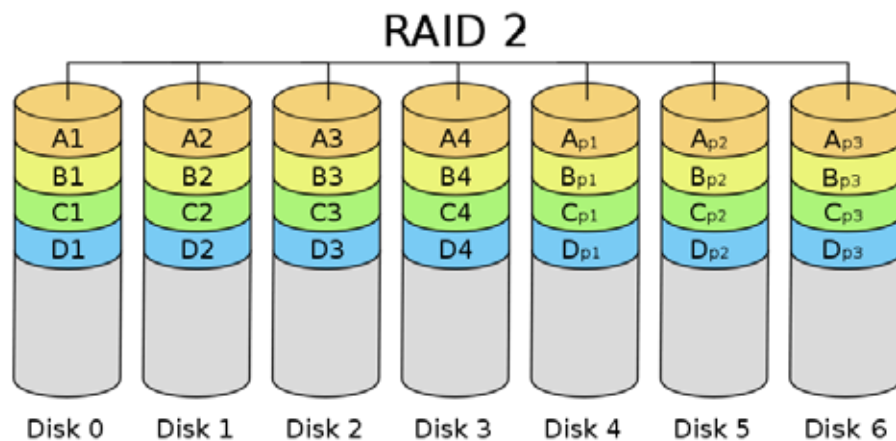
Gambar 2.2. RAID 1

(Wikipedia online)

3. **RAID 2:** Juga menggunakan sistem *stripping*. Namun ditambahkan tiga *harddisk* lagi untuk pariti *hamming*, sehingga data menjadi lebih *reliable*. Karena itu, jumlah *harddisk* yang dibutuhkan adalah minimal 5 ($n+3$, $n>1$). Ketiga *harddisk* terakhir digunakan untuk menyimpan *hamming code* dari hasil perhitungan tiap *bit-bit* yang ada di *harddisk* lainnya.

Contoh:

Kita memiliki 5 *harddisk* (sebut saja *harddisk* A,B,C, D, dan E) dengan ukuran yang sama, masing-masing 40GB. Jika kita mengkonfigurasi keempat *harddisk* tersebut dengan RAID 2, maka kapasitas yang didapat adalah: $2 \times 40\text{GB} = 80\text{GB}$ (dari *harddisk* A dan B). Sedangkan *harddisk* C, D, dan E tidak digunakan untuk penyimpanan data, melainkan hanya untuk menyimpan informasi pariti *hamming* dari dua *harddisk* lainnya: A, dan B. Ketika terjadi kerusakan fisik pada salah satu *harddisk* utama (A atau B), maka data tetap dapat dibaca dengan memperhitungkan pariti kode *hamming* yang ada di *harddisk* C, D, dan E.



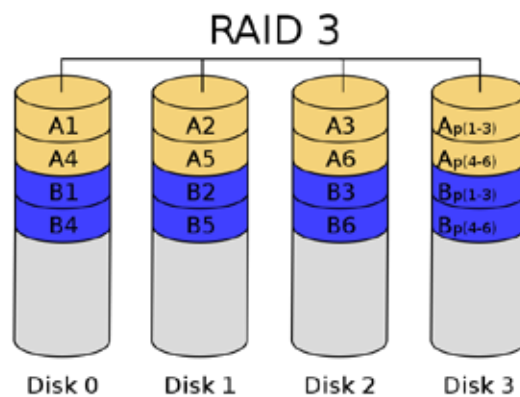
Gambar 2.3. RAID 2

(Wikipedia online)

4. **RAID 3:** RAID 3, juga menggunakan sistem *stripping*. Juga menggunakan *harddisk* tambahan untuk *reliability*, namun hanya ditambahkan sebuah *harddisk* lagi untuk *parity*. Karena itu, jumlah *harddisk* yang dibutuhkan adalah minimal 3 ($n+1$; $n > 1$). *Harddisk* terakhir digunakan untuk menyimpan pariti dari hasil perhitungan tiap *bit-bit* yang ada di *harddisk* lainnya.

Contoh kasus:

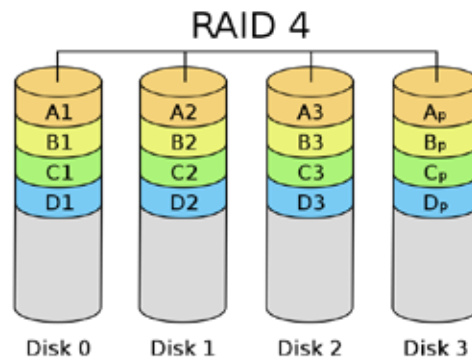
Kita memiliki 4 *harddisk* (sebut saja *harddisk* A,B,C, dan D) dengan ukuran yang sama, masing-masing 40GB. Jika kita mengkonfigurasi keempat *harddisk* tersebut dengan RAID 3, maka kapasitas yang didapat adalah: $3 \times 40\text{GB} = 120\text{GB}$. Sedangkan *harddisk* D tidak digunakan untuk penyimpanan data, melainkan hanya untuk menyimpan informasi pariti dari ketiga *harddisk* lainnya: A, B, dan C. Ketika terjadi kerusakan fisik pada salah satu *harddisk* utama (A, B, atau C), maka data tetap dapat dibaca dengan memperhitungkan pariti yang ada di *harddisk* D. Namun, jika *harddisk* D yang mengalami kerusakan, maka data tetap dapat dibaca dari ketiga *harddisk* lainnya.



Gambar 2.4. RAID 3

(Wikipedia online)

Kebutuhan *harddisk* minimalnya juga sama, $3(n+1 ; n > 1)$.

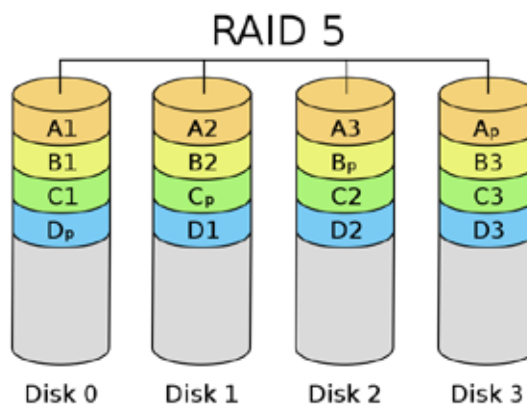


Gambar 2.5. RAID 4

(Wikipedia online)

5. **RAID 5** : RAID 5 pada dasarnya sama dengan RAID 4, namun dengan pariti yang terdistribusi. Yakni, tidak menggunakan *harddisk* khusus untuk menyimpan paritinya, namun paritinya tersebut disebar ke seluruh *harddisk*. Kebutuhan *harddisk* minimalnya juga sama, 3 ($n+1$; $n > 1$).

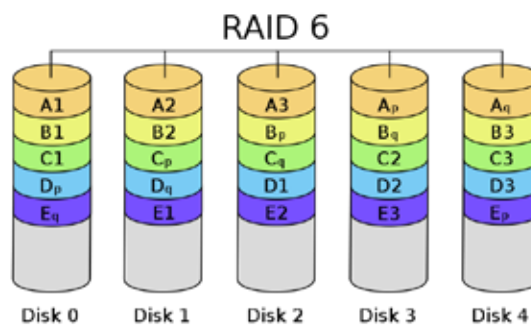
Hal ini dilakukan untuk mempercepat akses dan menghindari *bottleneck* yang terjadi karena akses *harddisk* tidak terfokus kepada kumpulan *harddisk* yang berisi data saja.



Gambar 2.6. RAID 5

(Wikipedia online)

- 6. RAID 6 :** Secara umum adalah peningkatan dari RAID 5, yakni dengan penambahan pariti menjadi 2 ($p+q$). Sehingga jumlah *harddisk* minimalnya adalah 4 ($n+2$; $n > 1$). Dengan adanya penambahan pariti sekunder ini, maka kerusakan dua buah *harddisk* pada saat yang bersamaan masih dapat ditoleransi. Misalnya jika sebuah *harddisk* mengalami kerusakan, saat proses pertukaran *harddisk* tersebut terjadi kerusakan lagi di salah satu *harddisk* yang lain, maka hal ini masih dapat ditoleransi dan tidak mengakibatkan kerusakan data di *harddisk* bersistem RAID 6.



Gambar 2.7. RAID 6

(Wikipedia online)

- 7. RAID 7 :** RAID7 membuat sistem operasi sebagai *controller*, *caching* menggunakan jalur cepat.

2.3 DRBD (Distributed Replicated Block Device)

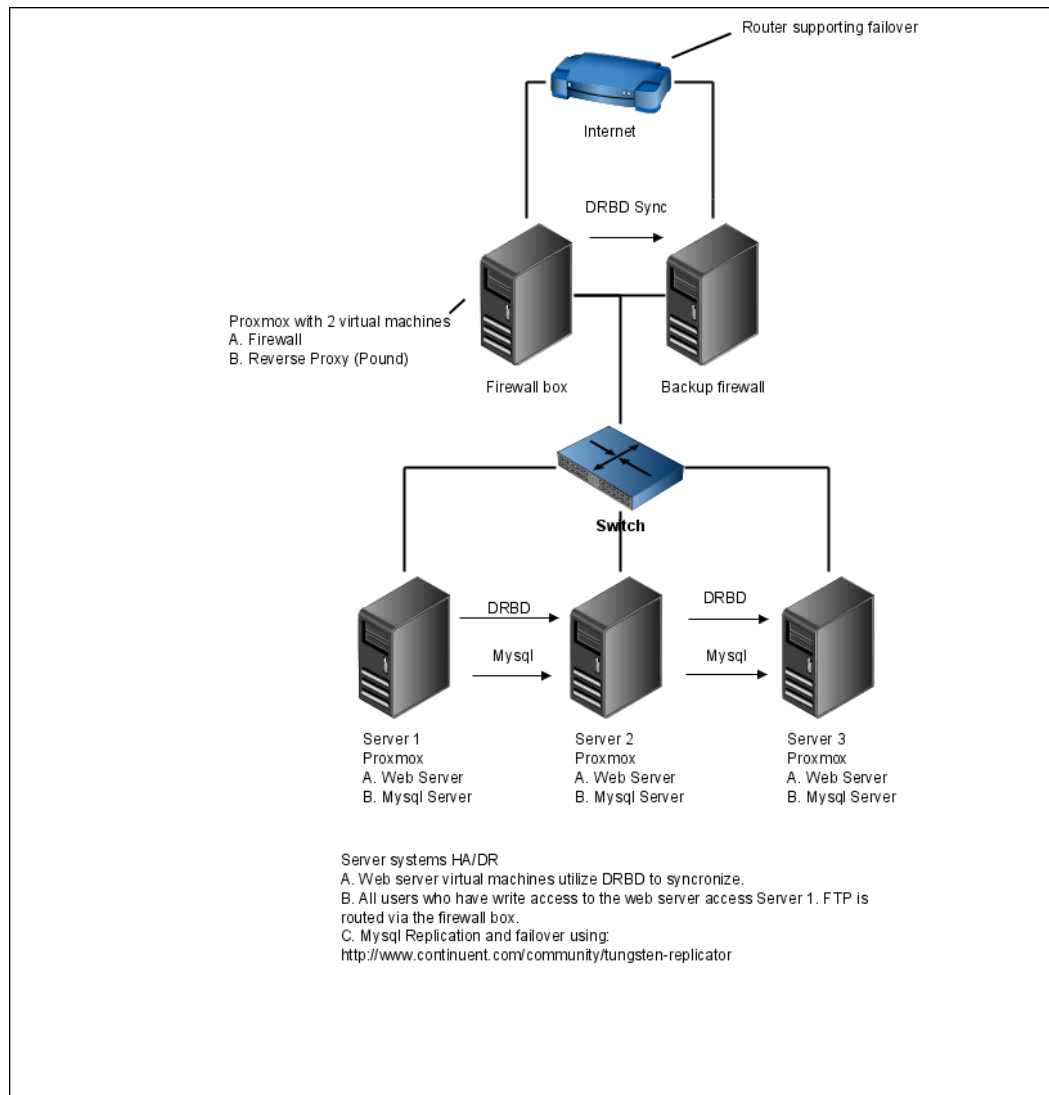
DRBD® bisa dianalogikan sebagai mekanisme RAID-1 (*mirroring*, bisa juga tipe RAID lain yang menggunakan prinsip *mirroring*), yang melakukan duplikasi data melalui *network*. Duplikasi data ini dilakukan

dalam mekanisme block devices, bukan dalam bentuk data mentah. Jika RAID-1 melakukan duplikasi isi dan data suatu *harddisk* atau partisi ke *harddisk* atau partisi lain, DRBD melakukan hal yang sama, hanya saja dilakukan melalui *network*.

DRBD dan *harddisk* RAID bersifat saling mendukung. DRBD memiliki satu keunggulan dibandingkan *harddisk* RAID, yaitu *backup server* berada terpisah dengan sumber *backup*. Pemisahan ini membawa keuntungan *preventif*, jika ada masalah pada salah satu *server*, *server* lainnya akan bertindak sebagai *server* pengganti. Jika *server* utama sudah kembali pulih, kendali akan dikembalikan ke *server* utama.

Fungsi DRBD sangat fleksibel untuk di terapkan pada berbagai macam fungsi layanan *server*. Beberapa contoh adalah *server web*, *server database*, *server file*. Bahkan *firewall* dan *proxy* sekalipun dapat menggunakan sistem DRBD untuk melakukan *backup* dan *failover*.

Berikut adalah gambar topologi DRBD yang menjadi mesin duplikasi untuk *server*, dalam hal ini adalah *server database* dan *server web*. Selain itu DRBD pada gambar juga menjadi mesin duplikasi untuk sistem *firewall*.



Gambar 2.8. DRBD

(Sumber : forum.proxmox.com)

Pada dasarnya DRBD adalah *software* untuk sinkronisasi data tetapi pada *level block device*, yaitu *level* dibawah *file system*. Jadi membuat *file system*-pun akan disinkronkan. Sinkronisasi pada *level block device* ini diperlukan (tetapi tidak harus) agar duplikasi bisa lebih lancar pergantian komputernya.

2.4 Sejarah dan Pengertian Linux Opensuse

SUSE sebelumnya bernama *SUSE Linux* dan *SuSE Linux Professional*, adalah salah satu distro Linux dari perusahaan Novell, atau lebih tepat dari anak perusahaannya Suse Linux GmbH (*Software-und System-Entwicklungsgesellschaft mbH, Nürnberg* yang berarti pengembangan perangkat lunak dan sistem).

SUSE Linux awalnya merupakan distro Slackware terjemahan bahasa Jerman. Ada informasi tidak resmi yang mengatakan bahwa S.u.S.E berhubungan dengan ilmuwan komputer Jerman Konrad Zuse karena pengucapan namanya yang sama.

Terdapat dua (2) distro utama *SUSE Linux* yang saat ini aktif:

1. ***SUSE Linux Enterprise*** : *SLE* adalah solusi sumber terbuka dari Novell untuk perusahaan besar. Terdiri dari dua paket, yaitu:
 - *SUSE Linux Enterprise Server* : SLES adalah Sistem Operasi *Server* Perusahaan yang merupakan komponen SLE.
 - *SUSE Linux Enterprise Desktop* : SLED adalah Sistem Operasi Desktop Perusahaan yang merupakan komponen SLE.
2. ***OpenSUSE*** : sebuah proyek masyarakat, yang disponsori oleh Novell, dirancang untuk pengguna rumahan.

SUSE bermula di awal tahun 1990-an di mana Linux terdiri dari sekitar 50 keping disket dan dapat diunduh/diambil lewat *internet*, tetapi pengguna potensial yang memiliki koneksi internet tidaklah banyak.

Kemudian S.u.S.E. GmbH menghimpun disket-disket Linux yang dapat dibeli (tanpa harus memiliki koneksi internet). SuSE tersebarluas oleh Suse GmbH dengan lokalisasi instalasi dalam bahasa Jerman dan dengan itu menciptakan distribusi dari banyak pengguna berbahasa Jerman. Alat instalasi dari *Slackware* diganti dengan YaST hasil pengembangan Suse GmbH sendiri. Mulai April 1994 Paket Suse-Linux Versi 1.0 mulai menggunakan CD, tidak lagi dalam disket (yang sudah mencapai 70 keping).

Versi pertama yang berdiri sendiri terlepas dari *Slackware* diterbitkan pada Mei 1996 dengan nama *S.u.S.E. Linux, versi 4.2*. Penomoran 4.2 dalam versi ini diakibatkan dari diskusi panjang di mana penomoran versi 1.1 ditolak dan angka 42 lebih disukai karena merupakan "jawaban dari segala pertanyaan terhadap segala pertanyaan" (*Answer to Life, the Universe, and Everything*)” menurut roman karya Douglas Adams *The Hitchhiker's Guide to the Galaxy*. Pada versi ini untuk pertama kalinya, dalam distribusi dengan 3 CD, disertai sebuah *Live-Filesystem*.

Mulai dari versi 4.2 angka penjualan *Suse Linux* meningkat tajam. Pengguna profesional di pasar Linux menuntut produk yang sesuai, maka mulai versi 5 ditawarkan produk *SuSE Business Linux*. Konsep ini kemudian tetap dijual melalui *SUSE Linux Enterprise Server (SLES)*, yang boleh diperoleh di samping siklus rilis dan pembaruan yang panjang dengan dukungan tawaran dan pelatihan yang beragam.

Suse Linux yang sampai pada versi itu hanya mendukung platform Intel i386, pada versi 6.1 mulai juga mendukung platform DEC, Alpha AXP dan platform PowerPC pada versi 6.3. Kedua distribusi memiliki pengaruh penting bagi pengembangan kualitatif Distribusi *Suse Linux*. Pada perkembangan berikutnya tersedia juga versi *SuSE Linux* untuk sistem AMD Athlon 64, Intel Itanium dan IBM 390 (Z-Series).

Mulai versi 7.0 sampai dengan 9.1 tersedia dua versi *Suse Linux*: *Personal* dan *Professional*. Di samping itu tersedia juga versi bagi pelajar. Paket pembaruan dengan harga yang pantas untuk versi *Professional* juga tersedia tanpa cetakan buku pedoman administrasi. Pada 4 November 2003, Novell mengumumkan bahwa mereka akan mengakuisisi SuSE. Akuisisi ini diselesaikan pada Januari 2004. SuSE 9.1 merupakan versi pertama di bawah Novell. Salah satu perubahan yang terjadi adalah mulai Juni 2004, di samping instalasi melalui FTP, CD untuk instalasi dasar tersedia di internet. Juga pada edisi *Professional* dipasarkan dengan keping DVD kedua yang berisi perangkat lunak untuk sistem 64-Bit (AMD64 dan Intel 64) (versi 64-Bit SuSE 9.0 dijual terpisah). Pada April 2004 YaST ditempatkan di bawah Lisensi Publik Umum GNU. Pada 4 Agustus 2005, juru bicara dan direktur hubungan masyarakat Bruce Lowry mengumumkan bahwa pengembangan *SUSE Professional* akan lebih terbuka dan bersama dalam proyek komunitas *openSUSE* berupaya meraih perhatian yang lebih luas dari pengguna dan pengembang. Lebih terbuka dengan memungkinkan pengguna dan pengembang untuk menguji dan membantu mengembangkannya. Sebelumnya segala pengembangan dilakukan hanya oleh *SUSE* dan versi 10.0

adalah versi pertama dengan pengujian beta oleh publik. Sebagai bagian dari perubahan, akses ke *Server-YaST* menjadi pelengkap bagi pengguna *SUSE Linux*. Maskot dari *SUSE* secara umum dikenali sebagai gecko (tokek) dan disebut sebagai *Gecko*, namun sebenarnya adalah seekor kameleon (*Chamaeleonidae*).

2.5 VMware Workstation v7.1.1 Full Version

VMware Workstation 7 merupakan *software* handal untuk virtualisasi *desktop* atau pembuat perangkat lunak dan pengujian terhadap perangkat lunak profesional IT yang sudah ada dimana perusahaan dapat menjalankan sistem lebih dari satu Operasi (*multiple*) pada sebuah komputer. Pengguna dapat menjalankan Windows, Linux, NetWare atau Solaris x86 artikel baru terhubung sepenuhnya, mesin portabel yang virtual memerlukan *restart* atau *Hardisk* untuk mempartisi regular. VMware Workstation 7 memberikan performa yang sangat baik dan fitur terdepan yang pengubahan dan memori optimasi mampu untuk mengatur konfigurasi multi-tier . Fitur yang terdapat di *VMware* yaitu jaringan virtual Pembongkaran, foto hidup, *drag* dan *drop* dan berbagi folder, dukungan dan PXE Membuat Aplikasi VMware Workstation menjadi regular tidak kuat dan dapat dipisahkan bagi sistem TI pengembang dan *administrator*.

VMware Workstation 7 digunakan untuk membuat dan menjalankan mesin *virtual* atau lebih dari satu komputer atau laptop di desktop anda. Anda dapat mengubah komputer yang sudah ada menjadi sebuah mesin virtual VMware, Membuat mesin virtual atau operasi sistem

yang baru. Masing-masing komputer virtual menghadirkan semua kebutuhan yang digunakan sebuah komputer, termasuk prosesor, memori, harddisk, jaringan dan koneksi port untuk perangkat.

VMware Workstation 7 memudahkan anda untuk menggunakan mesin virtual anda untuk menjalankan Windows, Linux dan operating sistem lainnya dengan komputer yang sama. Anda dapat berpindah antara sistem operasi lain secara langsung hanya dengan klik *mouse*.

2.5.1. Fitur Utama Dari VMware Workstation 7

Fitur Utama Dari VMware Workstation 7 adalah :

1. Kebanyakan Advanced Platform Virtualisasi VMware Workstation Menyediakan, berkinerja tinggi, dan aman mesin *virtual platform* yang paling diandalkan. Hal ini menawarkan *host* luas dan *guest* sistem pendukung operasi, pengalaman pengguna terkaya, dan set fitur yang paling komprehensif.
2. Jalankan Sistem Operasi Lainnya Dengan dukungan lebih dari 200 sistem operasi termasuk Windows 7, Windows Server 2008 R2 dan lebih dari 20 versi lain Windows, bersama dengan Redhat, Ubuntu, OpenSuse dan 26 versi tambahan Linux Workstation. VMware memberikan dukungan tamu *platform* luas untuk menjalankan beberapa sistem operasi pada saat bersamaan pada komputer Anda.

BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1. Analisa

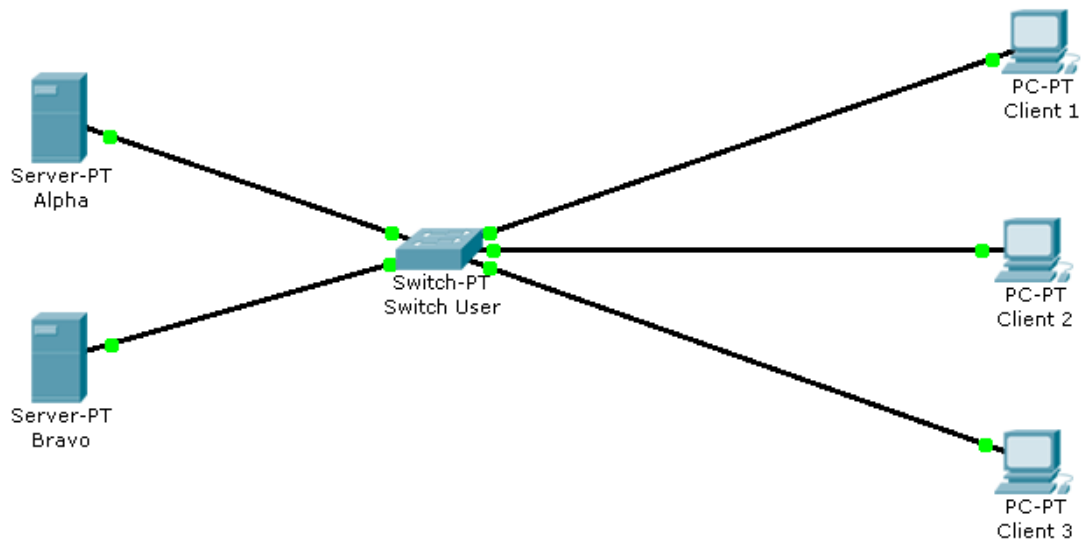
Pada dasarnya DRBD adalah *software* untuk sinkronisasi data tetapi pada *level block device*, yaitu *level* dibawah *file system*. Jadi membuat *file system*-pun akan disinkronkan. Sinkronisasi pada *level block device* ini diperlukan (tetapi tidak harus) agar HA dapat lebih lancar dalam penggantian ke *server* cadangan/duplikat.

Perancangan sistem duplikasi ini di tujukan untuk mempersingkat waktu yang di butuhkan oleh seorang *administrator* dalam melakukan *failover* pada sistem yang rusak. Untuk melakukan *failover* pada DRBD relative lebih cepat penanganannya dibandingkan dengan menggunakan RAID-1. Yang mana pada RAID-1 apabila kerusakan pada komponen komputer selain *harddisk*, misalnya *motherboard* maka akan dibutuhkan waktu yang lebih lama untuk melakukan penggantian komponen. Sedangkan apabila menggunakan DRBD administrator hanya akan melakukan *failover* dengan cara mengganti komputer yang rusak dengan komputer cadangan yang terkoneksi dengan sistem DRBD.

3.2. Perancangan Sistem

Perancangan sistem berisikan penjelasan tentang deskripsi umum sistem, proses-proses akan dijabarkan dalam topologi maupun beberapa permodelan yang terkait dengan sistem jaringan komputer serta *failover* pada *server*.

3.2.1. Deskripsi Umum Sistem



Gambar 3.1 Topologi DRBD

Deskripsi dari rancangan sistem tersebut adalah sebagai berikut :

Input :

Server DRBD akan menerima input dari Client berupa data untuk di simpan pada *hardisk* yang memiliki fungsi DRBD yang telah disiapkan pada *server Alpha*.

Proses :

Secara sistem data yang telah masuk pada *hardisk* DRBD pada *server Alpha* akan di duplikasikan pada *hardisk* DRBD yang ada pada *Bravo*.

Output :

Keluaran yang di hasilkan oleh sistem *failover* ini adalah sebuah duplikat dari fungsi server utama yang sama hasil dari data yang telah di proses pada user.

3.2.2. Kebutuhan Pengguna (*User*)

Dalam kebutuhan pengguna (*user*) pada sistem *failover* ini hanyalah bila *user* tersebut adalah seorang *administrator*. Hanya *administrator* saja yang dapat melakukan modifikasi dan melakukan konfigurasi pada sistem *failover* ini. Misalnya apabila dilakukan penambahan *harddisk* untuk kebutuhan DRBD yang lebih besar.

3.2.3 Kebutuhan Sistem

Dalam pembangunan sistem pengamanan jaringan ini di perlukan berbagai macam komponen, baik yang bersifat wajib, maupun tambahan atau pelengkap.

Untuk memenuhi kebutuhan pengguna (*user*) mengenai interaksi dengan sistem dan untuk mengetahui kebutuhan-kebutuhan apa saja yang perlu dipersiapkan, maka perlu dijabarkan mengenai kebutuhan baik dari sudut pandang perangkat keras maupun perangkat lunak.

a. Perangkat keras dalam hal ini adalah :

1. *Personal Computer* maupun *Server*
2. *Switch*
3. *Network Interface Card*
4. *Modem*, jika sistem jaringan ingin juga terkoneksi ke jaringan publik (*Internet*).
5. *Wireless Access Point (Non Router)*, bila menginginkan koneksi nirkabel.

b. Perangkat lunak dalam hal ini adalah :

1. Sistem Operasi linux opensuse versi 11.2.
2. *Tool* pendukung (DRBD system dan heartbeat)

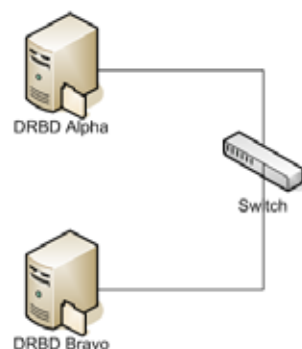
3.3. Alur Kerja DRBD Failover

DRBD® bisa dianalogikan sebagai mekanisme RAID-1 (*mirroring*, bisa juga tipe RAID lain yang menggunakan prinsip *mirroring*), yang melakukan duplikasi data melalui *network*. Duplikasi data ini dilakukan dalam mekanisme *block devices*, bukan dalam bentuk data mentah. Jika RAID-1 melakukan duplikasi isi dan data suatu *harddisk* atau partisi ke *harddisk* atau partisi lain, DRBD melakukan hal yang sama, hanya saja dilakukan melalui *network*.

DRBD dan *harddisk* RAID bersifat saling mendukung. DRBD memiliki satu keunggulan dibandingkan *harddisk* RAID, yaitu *backup server* berada terpisah dengan sumber *backup*. Pemisahan ini membawa keuntungan *preventif*, jika ada masalah pada salah satu *server*, *server* lainnya akan bertindak sebagai *server* pengganti. Jika *server* utama sudah kembali pulih, kendali akan dikembalikan ke *server* utama.

Teknik yang digunakan adalah dengan cara membangun dua buah *server* yang dihubungkan dengan metode DRBD. Artinya dua buah *server* ini benar – benar sama persis dalam hal isi dari *hardisk*, mengingat *server* kedua secara sistem adalah duplikat dari *server* pertama.

Dengan memanfaatkan sistem DRBD ini diharapkan waktu *downtime* pada sebuah *server* dapat diminimalisir.



Gambar 3.2. Analogi DRBD

Keterangan gambar 3.2 :

1. Pada saat HOST Alpha mati maka DRBD akan memerintahkan HOST Bravo untuk melakukan backup melalui bantuan *switch*.
2. Dan apabila HOST Alpha sudah kondisi hidup maka DRBD HOST Bravo akan memerintahkan host Alpha melakukan *backup* melalui bantuan *switch*.

BAB IV

IMPLEMENTASI SISTEM

4.1. Kebutuhan Sistem

Sebelum menjalankan sistem *failover* melalui jaringan ini, ada beberapa hal yang perlu diperhatikan, antara lain kebutuhan sistem akan perangkat keras (*hardware*) dan perangkat lunak (*software*), serta langkah-langkah yang harus dilakukan untuk dapat melakukan instalasi aplikasi agar dapat berfungsi sebagaimana mestinya.

Pada bagian ini akan dijelaskan mengenai spesifikasi minimum *Personal Computer* maupun *Server* yang akan digunakan dalam implementasi sistem kali ini.

a. Perangkat keras dalam hal ini adalah :

1. *Personal Computer* maupun *Server*. Perangkat ini digunakan untuk media penanaman sistem pengamanan jaringan. Ketentuan minimum perangkat keras yang terkandung di dalamnya adalah :
 - (1) *Processor* P4 1,0 Ghz atau *AMD* setara (Untuk *Linux Non GUI*).
 - (2) *Processor Dual Core Dedicated* (Untuk *Linux* dengan *GUI*).
 - (3) Satu buah *Network Interface Card* (bila sistem *failover* ini dilakukan bukan melalui *VMware*).
 - (4) *RAM dual channel* 1 GB (bila sistem berdiri sendiri).
 - (5) *RAM dual channel* 2 GB (bila sistem menggunakan *VMware*).
 - (6) *Hardisk* 160 GB.
2. *Switch* dengan jumlah *port* sesuai kebutuhan.

3. *Modem*, jika sistem jaringan ingin juga terkoneksi ke jaringan publik (*Internet*).
4. *Wireless Access Point (Non Router)*, bila menginginkan koneksi nirkabel.

b. Perangkat lunak dalam hal ini adalah :

1. Sistem Operasi. Dalam Tugas Akhir ini penulis menggunakan Sistem Operasi *OpenSuse* versi 11.2. Untuk sistem operasi, disini sangat menentukan karena sistem *failover* ini akan digunakan untuk menduplikasi data yang di simpan pada *server*. Dalam Tugas Akhir ini penulis menggunakan *Opensuse* dikarenakan Linux keturunan Redhat ini memang didedikasikan untuk digunakan sebagai *server*, sehingga dengan menggunakan *Opensuse* penulis bisa menggabungkan berbagai fungsi *server* (*Web server*, *Samba*, *FTP*, dan lain- lain) sehingga penyimpanan dari *server-server* tersebut dapat penulis buat duplikatnya.

4.2. Aplikasi Sistem Failover

Pada tahap ini akan dibahas mengenai implementasi DRBD dari perancangan yang telah dibahas sebelumnya.

DRBD adalah suatu sistem yang bisa dianalogikan sebagai mekanisme RAID-1 (*mirroring*, bisa juga tipe RAID lain yang menggunakan prinsip *mirroring*), yang melakukan duplikasi data melalui *network*. Duplikasi data ini dilakukan dalam mekanisme *block devices*, bukan dalam bentuk data mentah. Jika RAID-1 melakukan duplikasi isi dan data suatu *harddisk* atau partisi ke *harddisk* atau partisi lain, DRBD melakukan hal yang sama, hanya saja dilakukan melalui *network*.

4.3. Konfigurasi komputer

Proses instalasi *Linux opensuse 11.2* :

1. *Burning* file iso menjadi DVD, file iso *instalasi openSUSE* dapat didownload pada beberapa alamat berikut ini :

<http://kambing.ui.ac.id/iso/opensuse/11.2/>

<http://mirror1.opensuse.or.id/distribution/11.2/iso/>
2. Pilih versi DVD yang sesuai dengan arsitektur *processor* yang digunakan. Sebagai catatan, file yang mengandung tulisan *ix86* (i386, i486, i586, i686) berarti diperuntukkan bagi *processor 32 bit* sedangkan untuk file yang mengandung tulisan *x86_64* diperuntukkan bagi komputer yang memiliki *processor 64 bit*.
3. *Burning file iso* tersebut menggunakan aplikasi *burner DVD*, misalnya *Nero Burning ROM* pada sistem Windows atau *K3B* atau *Brasero* pada sistem Linux.
4. *Boot* komputer dengan posisi *CDROM/DVD ROM* sebagai pilihan pertama pada *boot device priority* di *BIOS*



Gambar 4.1 Pemilihan Menu Installation

5. Pilih *Installation*. Pada pilihan ini, penulis dapat mengubah pilihan bahasa untuk instalasi, ukuran layar, lokasi sumber *instalasi* (DVD atau

melalui *network*), pilihan kernel dan tambahan *driver*. Karena menggunakan DVD, penulis bisa langsung memilih kondisi *default* dengan menekan tombol ENTER.

6. Kita akan masuk ke posisi *Welcome Screen* yang menampilkan *openSUSE License Agreement*. Perhatikan bahwa *openSUSE* tidak meminta penulis setuju atau tidak seperti halnya instalasi Windows. Ini karena *openSUSE* bersifat bebas dan gratis untuk dipergunakan. Tampilan *license agreement* berisi informasi *distro Linux* dan peran komunitas dalam proses pembuatannya. Kita bisa langsung memilih tombol *Next*.



Gambar 4.2 Pemilihan Bahasa OpenSuse

7. *OpenSUSE* akan melakukan analisa sistem, deteksi hardware dan spesifikasi sistem yang ada. Tunggu sebentar hingga proses deteksi selesai dilakukan.
8. Pilihan berikutnya adalah pilihan instalasi, apakah berupa Instalasi Baru, *Update* atau Perbaiki instalasi sebelumnya. Jika *harddisk* yang dipergunakan sudah memiliki sistem *openSUSE*, pilihan *update* atau *repair* secara otomatis akan diaktifkan. Karena proses ini merupakan instalasi awal, pilih *New Installation*.



Gambar 4.3 New Installation

9. Berikutnya adalah menentukan waktu dan area waktu (Time Zone). Pilih Asia Jakarta jika memang tinggal di daerah WIB dan pilih area lokasi lain jika tinggal di area waktu WITA/WIT.



Gambar 4.4 Time Zone

10. Pilihan berikutnya adalah menentukan desktop manager. Tersedia berbagai pilihan, baik Gnome, KDE 3.5, KDE 4 maupun yang lain (XFCE, minimal system dll). Sesuai dengan judul panduan ini, penulis memilih untuk menggunakan *Gnome Desktop Manager* sebagai *desktop manager*. Anda bisa memilih untuk menggunakan *Minimal Server Selection* jika memang sudah terbiasa dengan konfigurasi berbasis teks. Pada dasarnya, pemilihan *desktop manager* adalah preferensi pribadi masing-masing. Anda bebas dan

boleh memilih *desktop manager* yang paling nyaman untuk dipergunakan. Secara prinsip perbedaan masing-masing desktop manager terletak pada tampilan dan sistem menu. Bisa saja aplikasi KDE dijalankan pada desktop manager Gnome atau sebaliknya.



Gambar 4.5 Desktop Manager

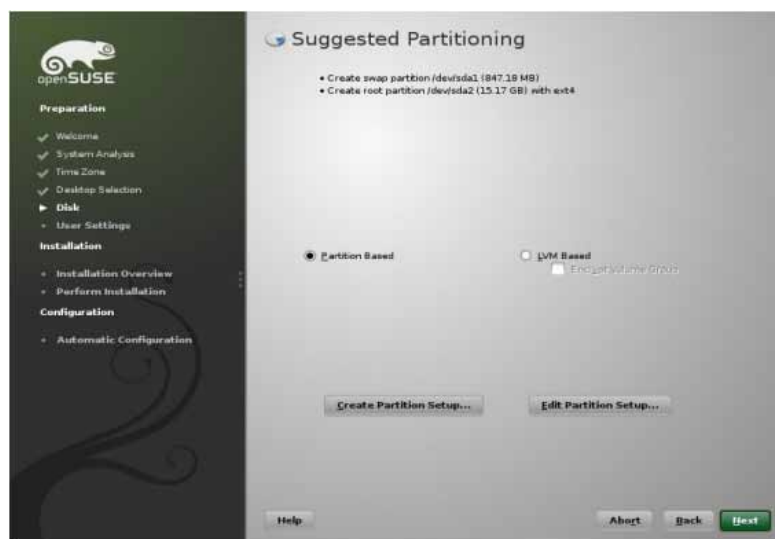
11. Setelah menentukan *Desktop Manager*, tahap selanjutnya adalah menentukan formasi *harddisk*. Jika sudah memiliki data pada *harddisk*, tahap ini adalah tahap yang perlu perhatian *extra* agar jangan sampai data yang sudah ada termasuk kedalam bagian yang akan diformat. *openSUSE* memiliki kemampuan mendeteksi partisi Windows dan partisi Linux lainnya dan menjaga agar data tersebut tidak hilang. Meski demikian, sangat disarankan untuk melakukan *backup* data terlebih dahulu secara *default* biasanya *openSUSE* memberikan kapasitas besar untuk */home*. Sebaiknya ubah formasi ini dengan cara memilih partisi/(*partisi root*) yang lebih besar daripada */home* karena akan menjadi folder utama dan melakukan *resize* ukuran/jauh lebih sulit daripada melakukan *resize* ukuran partisi lain.

Sebagai contoh, untuk *harddisk* 80 GB, penulis akan memilih 50 GB untuk *root* (/), 1.5 GB untuk *Swap* dan sisanya untuk */home*. Jika sistem sudah didedikasikan secara khusus untuk aplikasi tertentu, misalnya untuk *Zimbra mail server* yang menyimpan semua data pada folder */opt*, penulis tidak perlu membuat partisi */home* secara terpisah karena yang dibutuhkan justru partisi */opt*. Jika penulis tidak membuat suatu partisi secara terpisah, semua partisi akan diletakkan didalam *folder*. Jika menggunakan kapasitas *harddisk* terbatas, misalnya hanya 20 atau 40 GB, penulis lebih menyarankan untuk hanya membuat partisi dan swap, karena kalau dipisah kedalam beberapa partisi akan mengurangi kapasitas maksimum masing-masing partisi.

- a. Ada beberapa saran yang menganjurkan untuk memisahkan partisi sistem tertentu (misalnya */var*, */srv* dll) kedalam partisi terpisah. Jika memiliki *harddisk* yang berbeda, saran ini bisa diadopsi untuk meningkatkan kecepatan proses, namun jika menggunakan *harddisk* yang sama, penulis lebih menyarankan untuk tetap memilih formasi seperti diatas.
- b. Banyak juga yang menyarankan ukuran swap 2.5X memori fisik. Hal ini tidak sepenuhnya benar karena swap sebenarnya hanya *memory buffer* atau cadangan dan tidak akan digunakan jika memori fisik sudah cukup besar. Penggunaan swap juga akan memperlambat proses sistem sehingga *swap* merupakan cadangan darurat saja. Jika *swap* terpakai cukup besar, itu sudah merupakan tanda bahwa memori fisik harus ditingkatkan.
- c. Dalam banyak sistem, ukuran swap sebesar 1-1.5 GB untuk memory ≥ 1 GB sudah cukup sesuai sebagai antisipasi. Jika penulis memiliki 2

harddisk atau lebih, penulis bisa menerapkan sistem *raid* untuk perlindungan data. *Raid* adalah mekanisme penggunaan sistem yang menyatukan *harddisk* kedalam 1 *logical* partisi. Penjelasan lebih jauh mengenai *raid* akan dibahas dalam bab/artikel terpisah. Secara prinsip, jika penulis membuat partisi/dalam jumlah yang cukup, penulis bisa dengan mudah menambahkan kapasitas *harddisk* untuk dijadikan atau *dimount* sebagai partisi tertentu jika memang diperlukan.

12. Untuk mengubah formasi partisi, penulis bisa memilih menu *Edit Partition Setup*



Gambar 4.6 *Edit Partition Setup*

13. Sebelum proses instalasi dilakukan, ada *overview* mengenai pilihan yang sudah dilakukan. Kita masih bisa melakukan perubahan dari halaman *overview* ini. Halaman ini bisa dianalogikan sebagai *final confirmation*.
14. Kita akan mengurangi *software* yang di *install* karena sistem ini diperuntukkan sebagai server dan penulis tidak memerlukan aplikasi yang aneh-aneh (dalam arti, aplikasi multimedia tidak penulis perlukan di *server*).

15. Klik pada group *Software*.
16. Klik pada *Gnome Desktop Environment* hingga tanda centangnya hilang. Ini memastikan bahwa *Gnome Desktop* tidak akan di *install*.
17. Klik pada pilihan *Gnome Base System*. Ini akan memastikan bahwa penulis melakukan instalasi *Gnome basis*, bukan *Gnome Desktop full*. *Gnome basis* bisa menghemat hingga sepenulir 500 MB jika dibandingkan dengan *Gnome Desktop full*.
18. Klik pada *Novell AppArmor* hingga warna centangnya berubah dari warna hijau menjadi warna hitam. Ini berarti bahwa *Novell Apparmor* tidak akan diaktifkan namun tidak bisa semuanya dihilangkan karena ada paket *depedency*.
19. Klik OK
20. *Scroll* ke bagian paling bawah dari halaman *Installation Setting*. Klik *link Disable* pada tulisan "*Firewall will be enabled*". Untuk menghindari kemungkinan adanya *service* yang bermasalah dengan konfigurasi *setup firewall*, *firewall* akan dimatikan pada saat instalasi dan konfigurasi sampai nanti waktunya akan diaktifkan jika server sudah selesai *disetup* dan siap untuk dipergunakan sebagai *server production*.
21. Periksa sekali lagi apakah ada *setting* tertentu yang ingin diganti. Jika sudah OK semua, silakan klik tombol *Install*.



Gambar 4.7 Installation Setting



Gambar 4.8 Confirm Instalation

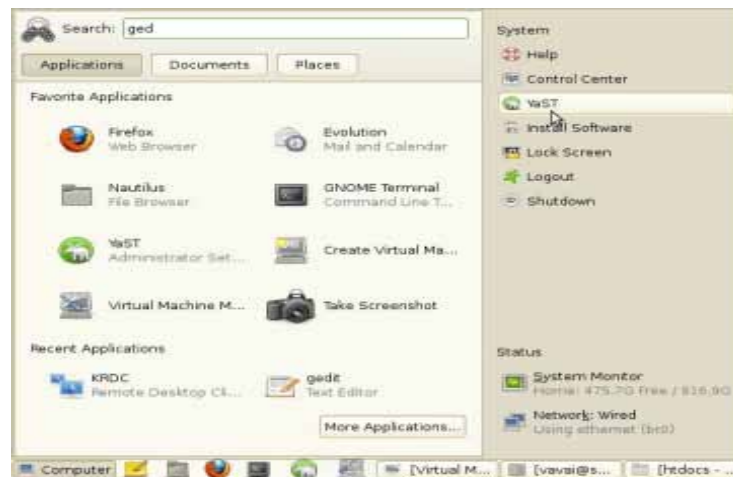
22. Tahap terakhir adalah proses konfigurasi secara otomatis untuk menentukan resolusi layar dan konfigurasi *hardware* lainnya.
23. Setelah proses ini selesai, penulis bisa menggunakan *openSUSE 11.2* dengan leluasa.

4.3.1 Konfigurasi network

Setelah *server* selesai diinstall, silahkan *login* dengan menggunakan *user* yang sudah didefinisikan pada saat instalasi. Jika pada saat instalasi memilih opsi “*auto login*”, posisi saat ini semestinya sudah *login* sebagai *user*.

Langkah pertama yang perlu dilakukan adalah setup konfigurasi jaringan. Untuk melakukannya, lakukan langkah sebagai berikut :

1. Klik START Menu (komputer) kemudian klik YAST. **YAST** adalah *tools* konfigurasi *openSUSE* yang sangat *powerful* yang akan banyak digunakan pada tutorial selanjutnya.



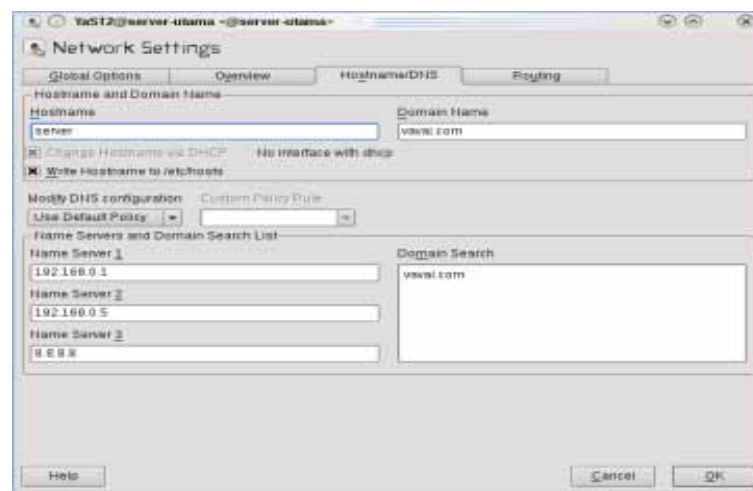
Gambar 4.9 YaST

2. Pilih Network Device | Network Setting



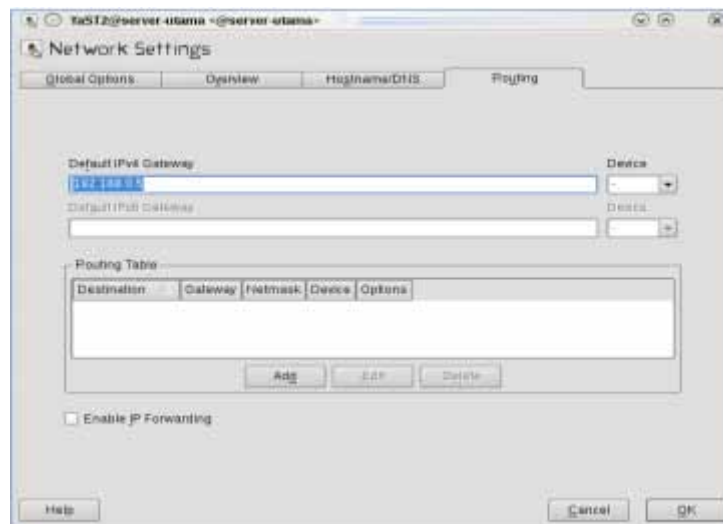
Gambar 4.10 Network Setting

3. Pilih network yang hendak dikonfigurasi, misalnya eth0, kemudian klik Edit
4. Masukkan IP Address dan Subnet mask. openSUSE mendukung penulisan subnetmask dalam format simple, misalnya 255.255.255.0 bisa ditulis /24. Hostname tidak usah diisi. Klik Next.
5. Pindah ke tab Hostname & DNS
6. Isi hostname, domain dan NS. Gunakan pola sebagai berikut :
 - a. Hostname : Nama komputer server, misalnya server
 - b. Domain : Fully Qualified Domain Name, misalnya virna.com
 - c. Name Server1 : IP Address DNS lokal. Bisa diisi dengan IP Address server jika server nantinya bertindak sebagai DNS Server
 - d. Name Server 2 : IP Address router/Modem ADSL
 - e. Name Server 3 : IP Address DNS ISP/Public DNS. 8.8.8.8 adalah IP Address public DNS milik Google



Gambar 4.11 Konfigurasi Host dan DNS

7. Pindah ke tab Routing
8. Isikan IP Address router/modem ADSL pada isian gateway. Dalam contoh modem ADSL memiliki IP 192.168.0.5



Gambar 4.12 Konfigurasi Gateway

9. Klik OK

Dengan formasi diatas, semestinya sudah bisa mengakses internet (jika memang akses).

4.4. Konfigurasi dan Instalasi DRBD

Berikut adalah langkah-langkah instalasi dan konfigurasi DRBD :

1. Jika menggunakan SUSE Linux Enterprise Server 11, download file iso SUSE® Linux Enterprise High Availability Extension dari

<http://www.novell.com/products/highavailability/>.

2. Download bisa dilakukan menggunakan free register account. Seperti halnya SLES 11, SUSE® Linux Enterprise High Availability Extension dapat digunakan tanpa biaya jika tidak membutuhkan support dan update dari Novell.

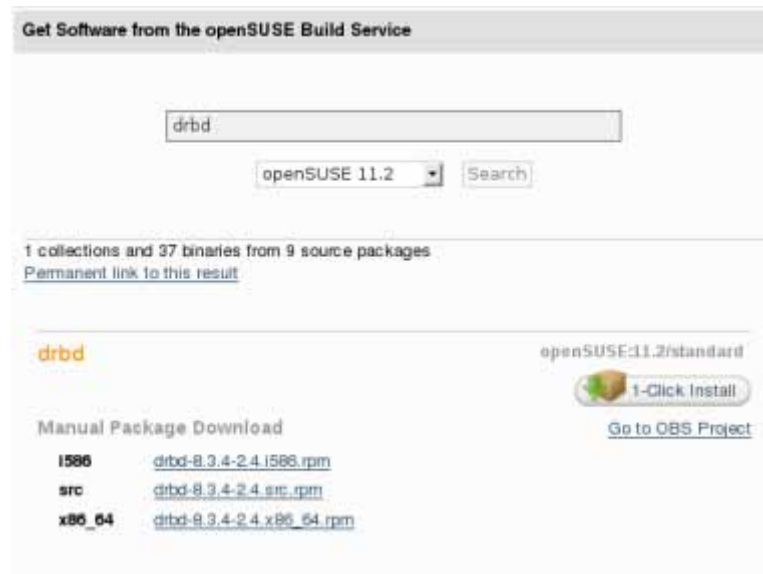
3. Pilih file iso sesuai arsitektur processor (misalnya penulis memilih iso versi x86_64 untuk server 64 bit penulis). Jika menggunakan openSUSE 11.2, DRBD bisa diinstal melalui repositories Linux HA

http://download.opensuse.org/repositories/server:/haclustering/openSUSE_11.1

4. atau melalui one-click-install.

Caranya :

- Buka <http://software.opensuse.org/search>
- Pilih openSUSE 11.2 dari daftar distro, kemudian ketikkan drbd pada kotak pencarian dan klik search



Gambar 4.13 One-Click Install

- Klik tombol 1-click-install pada sisi kanan, openSUSE akan secara otomatis melakukan instalasi paket aplikasi drbd melalui wizard
5. Install 2 buah sistem SLES 11 atau openSUSE 11.2. Penulis menggunakan nama alpha.virna.com untuk server pertama dan bravo.virna.com untuk server kedua. Nama ini akan menjadi rujukan konfigurasi DRBD. Silakan pilih apakah ingin menggunakan GUI atau text mode. Penulis menggunakan IP 192.168.1.20 untuk alpha.virna.com dan IP 192.168.1.21 untuk bravo.virna.com. Untuk instalasi kedua sistem tersebut, penulis menggunakan Xen Hypervisor. Proses instalasi juga dapat

dilakukan menggunakan VirtualBox atau VMWare atau KVM atau mekanisme virtual appliance lainnya.

6. Tambahkan harddisk virtual kedalam masing-masing sistem. Harddisk virtual ini nantinya akan menjadi device DRBD yang akan disinkronisasi via network. Ingat, jangan format/partisi harddisk ini karena partisi akan dilakukan setelah DRBD dikonfigurasi. Jadi total akan ada 2 harddisk virtual, yaitu harddisk pertama untuk sistem dan harddisk kedua untuk data drbd. Loncati instruksi nomor 4 dan 5 dibawah ini jika menggunakan openSUSE.
7. Jika Menggunakan SLES, Tambahkan Repo High Availability. Jika file iso yang didownload pada tahap pertama sudah diburn ke CD, gunakan CD tersebut sebagai addon repo dan aktifkan melalui YAST | Software | Software Repositories. Jika masih dalam bentuk ISO, tambahkan file iso tersebut sebagai media repo dengan cara yang sama, hanya saja memilih tipe “Local ISO” instead of CD/DVD.
8. Install Pattern “High Availability” Melalui YAST. Buka YAST | Software | Software Management dan berikan tanda centang pada pattern High Availability, kemudian klik Accept. Setelah selesai, tutup YAST.

Sebagai catatan, berikut adalah beberapa konfigurasi yang digunakan :

Konfigurasi Network Server :

```
alpha:~ # ifconfig
eth0    Link encap:Ethernet HWaddr 00:16:3E:13:14:40
inet addr:192.168.1.20 Bcast:192.168.1.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:57926 errors:0 dropped:0 overruns:0 frame:0
TX packets:605 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:8216139 (7.8 Mb) TX bytes:191596 (187.1 Kb)
```

```

lo      Link encap:Local Loopback
Inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)

```

```

alpha:~ # tail /etc/hosts
fe00::0      ipv6-localnet
ff00::0      ipv6-mcastprefix
ff02::1      ipv6-allnodes
ff02::2      ipv6-allrouters
ff02::3      ipv6-allhosts
127.0.0.2    alpha.virna.com alpha
192.168.1.21 bravo.virna.com bravo

```

```

alpha:~ # tail /etc/resolv.conf
# See also the netconfig(8) manual page and other
documentation.
#
# Note: Manual change of this file disables netconfig too, but
# may get lost when this file contains comments or empty lines
# only, the netconfig settings are same with settings in this
# file and in case of a "netconfig update -f" call.
#
### Please remove (at least) this line when you modify the file!
nameserver 192.168.10.2

```

Konfigurasi Harddisk Server

```

alpha:~ # fdisk -l
Disk /dev/sda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x0008f08c
Device Boot Start End Blocks Id System/dev/xvda1 1 206
1654663+ 82 Linux swap / Solaris/dev/xvda2 207 1305
8827717+ 83 Linux
Disk /dev/xvdb: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00000000
Disk /dev/sdb doesn't contain a valid partition table

```

Keterangan :

Disk `/dev/sdb` dinyatakan tidak memiliki partisi yang valid karena memang belum dibuatkan partisi. Penulis menggunakan kapasitas masing-masing sebesar 10 GB untuk harddisk virtual pertama maupun harddisk virtual kedua pastikan bahwa package `drbd`, `yast2-drbd` dan `drbd-kmp-default` atau `drbd-xen-default` (tergantung kernel yang digunakan oleh alpha maupun bravo) sudah terinstall dan server sudah direstart.

4.4.1 Konfigurasi DRBD

1. Ubah konfigurasi. Masuk ke konsole server Alpha dan buka file `/etc/drbd.conf`. File ini merupakan file konfigurasi utama DRBD. Isi file konfigurasi tersebut dengan isi sebagai berikut (jangan lupa, sesuaikan IP Address dan partisi harddisk) :

```

global {
    dialog-refresh 1;
    usage-count yes;
    minor-count 5;
}
common {
    syncer {
        rate 10M;
    }
}
resource r0 {
    protocol C;
    disk {
        on-io-error detach;
    }
    syncer {
        rate 10M;
    }
    al-extents 257;
}
on alpha {
    device /dev/drbd0;
    address 192.168.1.20:7788;
    meta-disk internal;
    disk /dev/sdb;

```

```

    }
on bravo {
    device /dev/drbd0;
    address 192.168.1.21:7788;
    meta-disk internal;
    disk /dev/sdb;
}
}

```

2. Untuk nama device menggunakan nama device standar untuk drbd yaitu /dev/drbd0. Untuk IP disesuaikan. Untuk harddisk juga disesuaikan partisi mana yang hendak di raid via network.
3. Salin file konfigurasi. Gunakan perintah scp untuk menyalin file /etc/drbd.conf dari server alpha ke server bravo. Letakkan pada foldernya yaitu /etc scp /etc/drbd.conf root@192.168.10.21:/etc/drbd.conf
4. Buat metadata disk dan jalankan service drbd.
5. Perintah ini dilakukan 2X, yaitu di server alpha dan kemudian di server bravo :

```

drbdadm create-md r0
service drbd start

```

6. Pesan yang muncul :

```

--= Thank you for participating in the global usage survey ==--
The server's response is:Writing meta data...
initialising activity log
NOT initialized bitmap
New drbd meta data block sucessfully created.Starting DRBD
resources:
[ d(r0) s(r0) n(r0) ].....

```

7. Check Status DRBD

```

service drbd status

```

8. Pesan yang muncul :

```
alpha:~ # service drbd status

drbd driver loaded OK; device status:
version: 8.2.7 (api:88/proto:86-88)
GIT-hash: a1b440e8b3011a1318d8bff1bb7edc763ef995b0
build by lmb@hermes,
2009-02-20 13:35:59
m:res cs st ds p mounted fstype
0:r0 Connected Secondary/Secondary
Inconsistent/Inconsistent C
```

9. Pesan diatas menunjukkan bahwa DRBD sudah berjalan, namun kedua server sama-sama bertindak sebagai secondary dan keduanya inconsistent (masing-masing node berbeda isi). Tak masalah, penulis akan mengaktifkannya sebentar lagi.

10. Jadikan server alpha sebagai primary (sumber utama proses sinkronisasi). Jalankan perintah berikut pada server alpha :

```
drbdsetup /dev/drbd0 primary --overwrite-data-of-peer
```

11. Check Ulang Status DRBD :

```
service drbd status
```

12. Pesan yang tampil :

```
alpha:~ # service drbd status
drbd driver loaded OK; device status:
version: 8.2.7 (api:88/proto:86-88)
GIT-hash: a1b440e8b3011a1318d8bff1bb7edc763ef995b0 build
by lmb@hermes, 2009-02-20 13:35:59
m:res cs st ds p mounted fstype
0:r0 SyncSource Primary/Secondary UpToDate/Inconsistent C...
sync'ed: 1.1% (10136/10236)M
```

13. Pesan diatas menunjukkan bahwa server alpha sudah berfungsi sebagai primary dan mulai melakukan sinkronisasi dengan server bravo (progress ditunjukkan dalam bentuk persentase).

14. Tunggu hingga proses sinkronisasi mencapai jumlah 100% dan kedua node sama-sama update. Cara mengeceknya adalah dengan perintah `service drbd status`. Lamanya waktu sinkronisasi tergantung pada besar kecilnya kapasitas disk yang menjadi node drbd dan kecepatan network antar kedua server.

Berikut adalah contoh status kedua node yang sudah sinkron :

```
alpha:~ # service drbd status
drbd driver loaded OK; device status:
version: 8.2.7 (api:88/proto:86-88)
GIT-hash: a1b440e8b3011a1318d8bff1bb7edc763ef995b0
build by lmb@hermes, 2009-02-20 13:35:59
m:res cs st ds p mounted fstype
0:r0 Connected Primary/Secondary UpToDate/UpToDate C
```

15. Setelah keduanya disinkronisasi penuh, penulis bisa membuat file system diatas node tersebut. Lakukan hal ini pada server alpha, tidak usah dilakukan diserver bravo karena server bravo akan otomatis melakukan duplikasi.

```
mkfs.ext3 /dev/drbd0
```

16. Tipe file sistem ext3 bisa diganti dengan *file system* lain yang didukung oleh kernel.

BAB VI

PENUTUP

6.1. Kesimpulan

Dalam implementasi pembangunan suatu sistem duplikasi dan *backup* serta pengujiannya dapat diambil beberapa kesimpulan yaitu :

1. Untuk membangun sebuah system backup serta duplikasi dibutuhkan suatu server yang didalamnya ditanam system DRBD. DRBD inilah yang nantinya akan menganalisa data apa saja yang masuk pada server utama dan melakukan backup danduplikasi pada server cadangan.
2. Pada kedua server telah ditanam informasi mengenai server kawannya, dari hasil informasi ini akan dapat di sinkronisasi antara kedua server. Sinkronisasi ini membutuhkan waktu, lamanya waktu sesuai dengan besar hardisk yang akan di sinkronisasikan. Makin besar hardisk yang akan digabungkan makin lama pula waktu yang dibutuhkan.
3. Untuk dapat melakukan suatu cara backup dan duplikasi *file* secara *realtime* pada server ini harus di berikan suatu system tambahan yakni *Heartbeat* atau *Open Ais* dan *Pace Maker*. Dengan system tambahan ini server diharapkan dapat mengenali apakah suatu server yang terhubung dengan dirinya mati atau hidup. Apabila dikenali suatu server utama yang terhubung dengan server cadangan dikenali mati atau non aktif oleh server cadangan maka server cadangan akan otomatis melakukan alih fungsi menggantikan posisi serta servis dari server utama.

BAB V

UJI COBA DAN EVALUASI

5.1. Lingkungan Uji Coba

Pada bab ini akan dibahas mengenai uji coba terhadap sistem Failover berbasis DRBD yang telah dibuat dan selanjutnya akan dibuat evaluasi dari hasil uji coba tersebut. Uji coba dilaksanakan untuk mengetahui apakah aplikasi dapat berjalan dengan baik sesuai perancangan yang dibuat. Evaluasi dilakukan untuk menentukan tingkat keberhasilan dari sistem yang dibuat.

Pada bagian ini akan dijelaskan mengenai spesifikasi *hardware* pada yang akan digunakan dalam implementasi sistem kali ini.

Pada uji coba kali ini digunakan sebuah *Personal Computer* sebagai *server/host* yang bertindak juga sebagai Sistem backup dengan spesifikasi sebagai berikut:

1. Sistem Operasi: *Opensuse* versi 11.2
2. Hardisk 20 GB tiap server
3. *Processor Intel Core 2 Duo Series*
4. *Ram* 1 GB

5.2. Skenario Uji Coba

5.2.1 Skenario Pengujian Langsung Pada Server

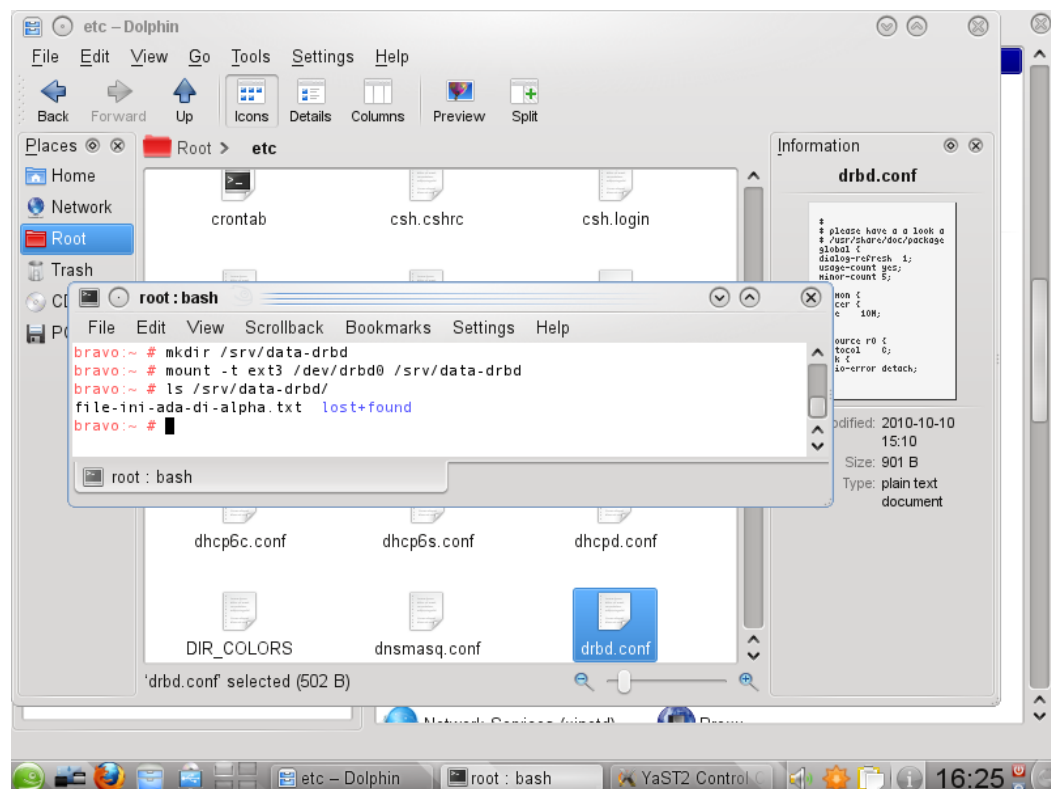
1. Mounting device DRBD pada server alpha, kemudian buat satu buah file sebagai testing. Nantinya file ini secara otomatis akan direplikasi/duplikasi ke server bravo. Ext3 bisa diganti dengan tipe partisi yang digunakan lain.


```

su
mkdir /srv/data-drbd
mount -t ext3 /dev/drbd0 /srv/data-drbd
touch /srv/data-drbd/ini-file-tesk-yang-ada-di-server-alpha.txt
ls /srv/data-drbd/

```

Perintah diatas akan berakhir dengan tampilan isi folder /srv/data-drbd yang berisi satu file teks. Pada prakteknya, data yang disimpan bisa dalam bentuk data apa saja.



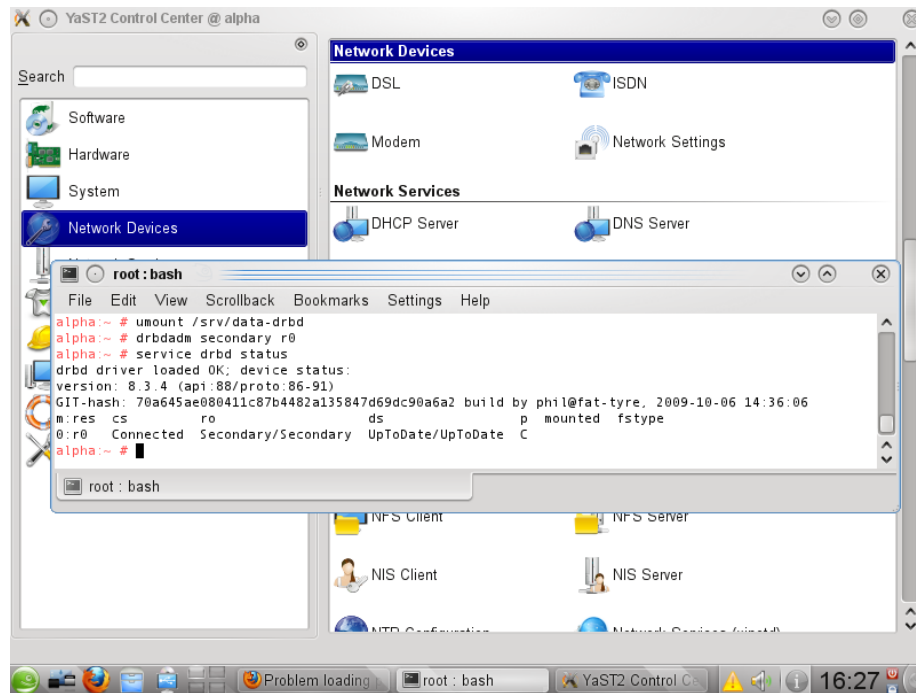
Gambar 5.1 Pembuatan sebuah file pada server Alpha

2. Unmount device DRBD dari server alpha, kemudian downgrade statusnya menjadi secondary

```

umount /srv/data-drbd
drbdadm secondary r0

```



Gambar 5.2 Downgrade status server Alpha

3. Pindah ke server bravo, aktifkan DRBD server bravo sebagai primary dan kemudian mounting device DRBD. Perhatikan bahwa server bravo secara otomatis akan memiliki salinan data dari server alpha

```

su
drbdadm primary r0
service drbd status
mkdir /srv/data-drbd
mount -t ext3 /dev/drbd0 /srv/data-drbd
ls /srv/data-drbd/

```

Kalau testing diatas sudah OK, kembalikan status DRBD ke kondisi semula yaitu server alpha sebagai primary dan server bravo sebagai secondary.

4. Pada server bravo :

```

umount /srv/data-drbd
drbdadm secondary r0

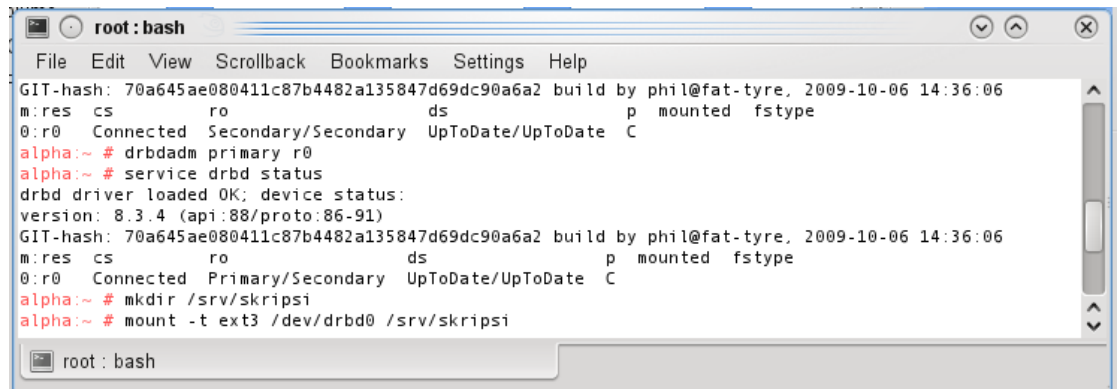
```

5. Pada server alpha :

```
drbdadm primary r0
service drbd status
mount -t ext3 /dev/drbd0 /srv/data-drbd
```

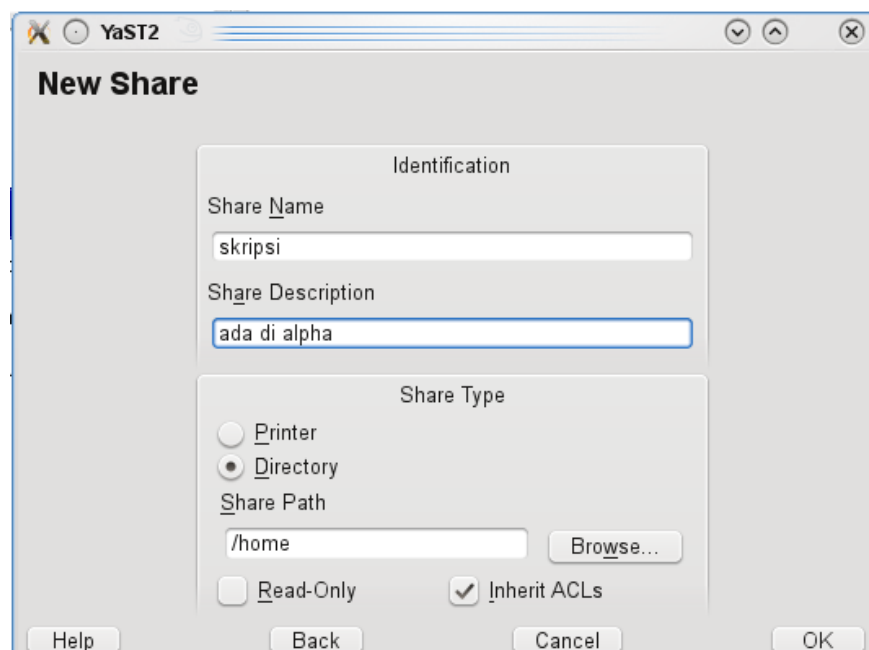
5.2.2 Pengujian Melalui Client (Skenario menggunakan Samba)

- Pada server Alpha di ciptakan sebuah folder “skripsi” yang nantinya digunakan sebagai DRBD sekaligus sebagai “file server” (samba)



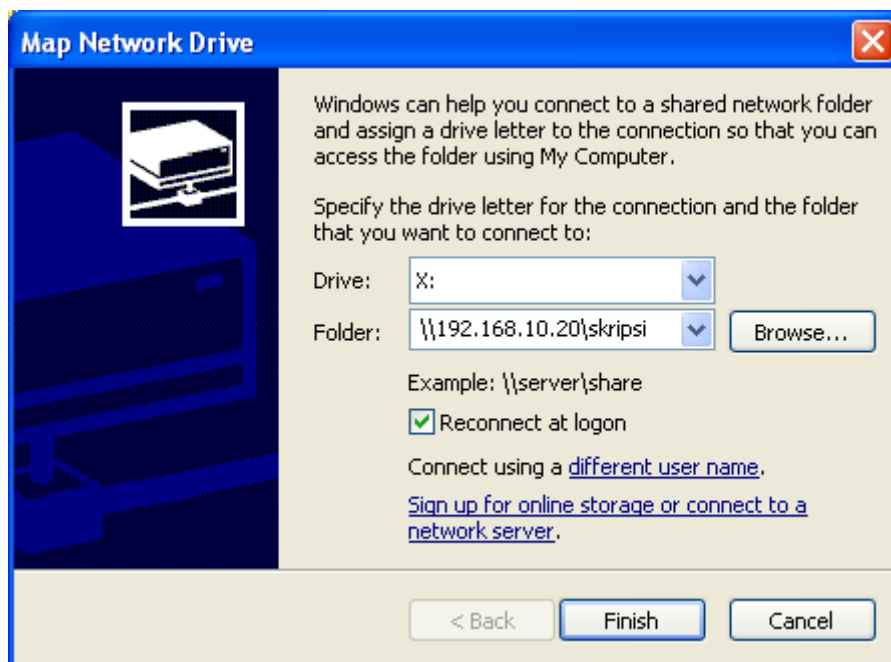
Gambar 5.3 Pembuatan direktori “skripsi”

- Mengaktifkan Server Samba dengan file sharing bernama “skripsi”



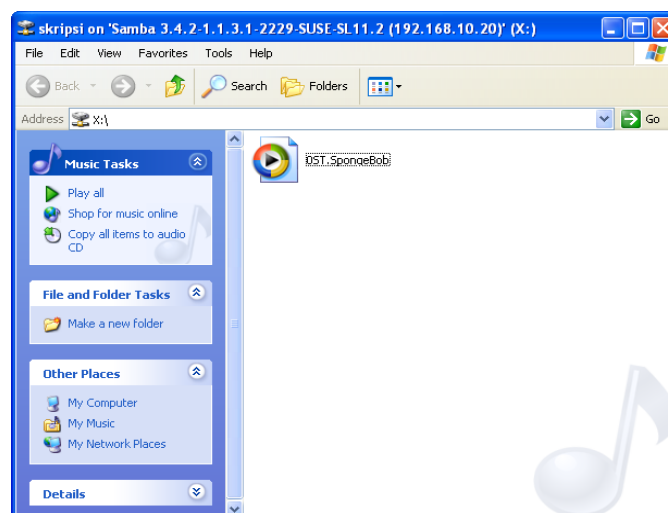
Gambar 5.4 Pembuatan File Sharing pada Alpha

- Pada Client (Windows), mencoba untuk mengakses Samba Server dengan melakukan “Map Network”



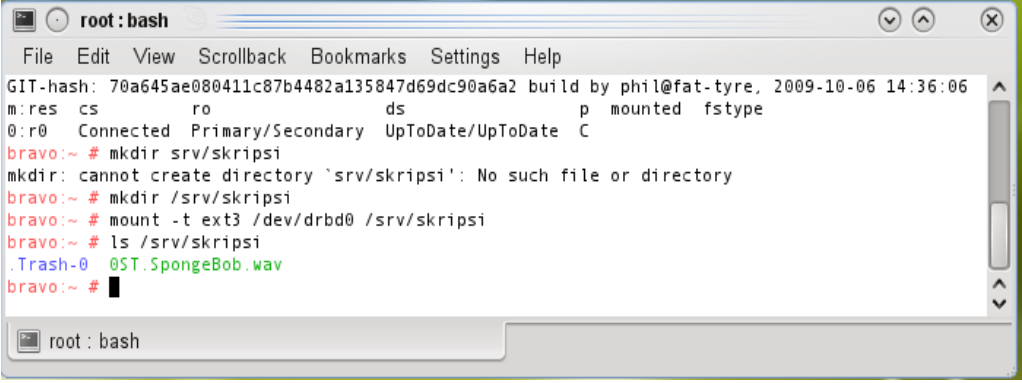
Gambar 5.5 Akses pada server Samba oleh client

- Client telah sukses masuk kedalam Server Samba. Kemudian client mengisi folder “skripsi” pada Server Samba dengan sebuah file



Gambar 5.6 Client mengisi sebuah file pada Samba

- Secara otomatis file yang tadinya oleh client di isikan pada Alpha telah terduplikasi pada Server Bravo



```
root: bash
File Edit View Scrollback Bookmarks Settings Help
GIT-hash: 70a645ae080411c87b4482a135847d69dc90a6a2 build by phil@fat-tyre, 2009-10-06 14:36:06
m:res cs ro ds p mounted fstype
0:r0 Connected Primary/Secondary UpToDate/UpToDate C
bravo:~ # mkdir srv/skripsi
mkdir: cannot create directory 'srv/skripsi': No such file or directory
bravo:~ # mkdir /srv/skripsi
bravo:~ # mount -t ext3 /dev/drbd0 /srv/skripsi
bravo:~ # ls /srv/skripsi
.Trash-0 @ST.SpongeBob.wav
bravo:~ #
```

Gambar 5.7 File telah terduplikasi pada server Bravo

6.2. Saran

Ada beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut pada sistem backup serta duplikasi berbasis jaringan menggunakan DRBD ini. Saran-saran itu adalah:

1. DRBD akan lebih dirasakan fungsinya bila administrator jaringan menanamkan fungsi *realtime failover* dengan menggunakan *heartbeat* atau *openais* dan *pacemaker*. Kemudahan yang didapat nantinya adalah administrator hanya bertindak mengawasi jalannya sistem saja mengingat proses *failover* dilakukan secara otomatis oleh mesin.
2. Menurut penelitian yang sudah ada, untuk sistem operasi lebih baik menggunakan Linux dari keluarga Red Hat karena lebih stabil serta dukungan untuk pengembangan lebih baik kedepannya (misalkan dalam modus virtualisasi berbasis linux)
3. Diharapkan untuk pengembangan kedepan sistem failover ini dapat di satukan dengan berbagai fungsi server, hingga diharapkan mungkin juga dilakukan melakukan fungsi failover pada router, proxy, serta firewall

DAFTAR PUSTAKA

- Barbakati, Naba, 1998. *Red Hat Linux Secrets 2nd edition*. IDG Book Worldwide, Inc. California.
- Binanto, Ilham, 2007. *Membangun Jaringan Komputer Praktis Sehari-hari*. Graha Ilmu. Yogyakarta.
- Proxmox. *DRBD-HA*, diakses *online* 25-08-2010. <http://forum.proxmox.com/threads/1188-Proxmox-VE-DRBD-HA-Storage>
- Sutedjo, Budi, 2004. *Konsep dan Perancangan Jaringan Komputer Bangunan Satu Lantai, Gedung Bertingkat dan Kawasan*. Andi Offset. Yogyakarta.
- Wikipedia. *Transmission Control Protocol/Internet Protocol*, diakses *online* 25-12-2009 pada http://id.wikipedia.org/wiki/Transmission_Control_Protocol/Internet_Protocol.